Android ROM boot

Vlad Victor Ungureanu

May 4, 2013

1. Overall process

In my proposal I have described how I would implement the booting procedure from an Android phone over ADK. This approach needed some previous files on the MMC. The ROM boot procedure needs the sys.boot pins to be changed as to have USB as 1^{st} selection. If a USB peripheral is detected the processor requests a file from that device(sending an ASIC code over the USB port). If the file is received is stored into the processors dedicated RAM(64KB) and ran. This first binary is a modified version of U-Boot [link]. After this binary is loaded into memory we need to download the full U-Boot which has support for MMC and after that getting the kernel+fs to the board is simple. On the Android device we need to rewrite [omap3_usbload] using the libusb that internally uses the Android USB API. When the final version of U-Boot is running on the board we can use the *loadb* command and with kermit send over the kernel+fs. The problem with kermit is that it requires root privileges on the Android device. I had a small talk with ka6sox about kermit and he told me that it should not be that hard to implement the protocol using libusb.

2. Issues

The main issue in this implementation is porting kermit on Android without the root permission needed. At this moment I do not know if I can boot over USB without the usage of kermit. If kermit can be avoided a generic send over usb tool can be used.

Not all devices support USB host mode so this approach is not available for all devices. For example the latest Nexus 4 does not support out of the box host mode(root+custom firmware can enable it). Here is a *almost* complete list of Android devices that support host mode [link].

3. Comparison between ROM boot and ADK boot

For ADK boot:

Pros	Cons
Does not need root access on the Android device.	Needs u-boot, small kernel+fs on the MMC or eMMC.
Provides an example project on how to use ADK to enable usage of Android device resources.	Final kernel and rootfs needs to be patched to ensure after booting connection with the Android device.
Uses ADK with is supported by every Android certified device.	Cannot be used as real debugging tool because everything depends on the MMC, eMMC.

For the ROM boot:

Pros	Cons
Boots from Android device without previous files from MMC, eMMC.	Maybe it needs rooted Android device(still researching).
Useful for debugging and board bring-ups.	Does not provide a example on how to use ADK to access the resource of the Android device.
Can boot a multitude of linux distros+rootfs(no need to patch rootfs to enable communication with Android device).	Not all Android devices support host mode.

4. Implementation plan

Time Frame	Milestone
12^{th} May - 27^{th} May	A part of my free time I will use to talk to my mentor and read on the technologies/tools I will have to use so I can hit the ground running from the first week of GSoC.
28^{th} May - 4^{th} Jun.	Use and understand pusb(or other USB download tool) with a PC and a BeagleBoard.
5^{th} Jun. -11^{th} Jun.	Compile and test libusb 1.0 with Android USB API support.
12^{th} Jun. - 18^{th} Jun.	Implement Android app to listen for ASIC codes.
19^{th} Jun. - 25^{th} Jun.	Test the code.
25^{th} Jun. - 3^{rd} Jul.	Implement kermit for Android with libusb that uses Android USB API(remove need of root privileges).
4^{th} Jul. - 11^{th} Jul.	Finish implementation and test the code.
12^{th} Jul. - 19^{th} Jul.	Midterm evaluation.
20^{th} Jul. - 27^{th} Jul.	Add app feature to download a kernel+rootfs(from internet) or take them from SD card and server them over USB.
28^{th} Jul. - 4^{th} Aug.	Test the code.
5^{th} Aug. - 12^{th} Aug.	Add option to either just debug the board without saving the kernel+rootfs on the MMC, eMMC or save them and use this procedure for first setup of a BeagleBoard.
13^{th} Aug. - 20^{th} Aug.	Test the code.
21^{st} Aug. - 28^{th} Aug	Test the full code on multiple Android devices(Nexus 7 tablet, Samsung Galaxy Nexus phone, Samsung Galaxy Fit phone, Samsung Galaxy Chat phone, Samsung Galaxy S, S II phones). Maybe more members of the community will test the bundle.
29^{th} Aug. - 5^{th} Sep.	Make code compatible with all the upper devices(if they have USB host) and solve the bugs that will arise.
6^{th} Sep. - 13^{th} Sep.	Find beta testers with different versions with Andoid on their devices to test the bundle and get feedback on how can the project can be improved.
14^{th} Sep. - 23^{rd} Sep.	Final testing, documentation, packaging and submitting the full work for final evaluation